# Reactive Motion Planning for Unmanned Aerial Surveillance of Risk-Sensitive Areas

Alex Wallar, Erion Plaku, and Donald A. Sofge

***Abstract*—This paper proposes a reactive motion-planning approach for persistent surveillance of risk-sensitive areas by a team of unmanned aerial vehicles (UAVs). The planner, termed PARCov (Planner for Autonomous Risk-sensitive Coverage), seeks to: i) maximize the area covered by sensors mounted on each UAV; ii) provide persistent surveillance; iii) maintain high sensor data quality; and iv) reduce detection risk. To achieve the stated objectives, PARCov combines into a cost function the detection risk with an uncertainty measure designed to keep track of the regions that have been surveyed and the times they were last surveyed. PARCov reduces the uncertainty and detection risk by moving each quadcopter toward a low-cost region in its vicinity. By reducing the uncertainty, PARCov is able to increase the coverage and provide persistent surveillance. Moreover, a nonlinear optimization formulation is used to determine the optimal altitude for flying each quadcopter in order to maximize the sensor data quality while minimizing risk.**

**The efficiency and scalability of PARCov is demonstrated in simulation using different risk models and an increasing number of UAVs to conduct risk-sensitive surveillance. Evidence of successful physical deployment is provided by experiments with AscTec Pelican quadcopters.**

***Note to Practitioners*—This paper was motivated by the viability of UAVs to enhance automation in environmental monitoring, search-and-rescue missions, package delivery, and many other applications. As UAVs become an economically feasible option for deployment, it becomes important to enhance their autonomy so as to increase productivity. In this paper, we develop an approach that uses simple interactions among UAVs to promote maximizing the area coverage while maintaining high sensor data quality and reducing the detection risk. The approach provides scalability, making it easy for UAVs to leave and join the mission as needed. Experimental results in simulation and with real quadcopters provide promising results. In future research, we would like to test and enhance the approach so that it can be used in various applications extending beyond laboratory testings.**

***Index Terms*—Motion planning, risk-sensitive surveillance, unmanned aerial vehicles.**

Manuscript received May 06, 2015; accepted June 04, 2015. Date of publication June 16, 2015; date of current version July 17, 2015. This paper was recommended for publication by Associate Editor D. Song and Editor Y. Sun upon evaluation of the reviewers' comments. This work was supported by the U.S. Department of Defense, Office of Naval Research under Grant N0001413WX21045, "Mobile Autonomous Teams for Navy Information Surveillance and Search (MANTISS)." The work of E. Plaku was supported by NSF IIS-1449505.

A. Wallar is with the School of Computer Science, University of St Andrews, Saint Andrews, Fife KY16 9AJ, Scotland, U.K. (e-mail: aw204@st-andrews.ac.uk).

E. Plaku is with the Department of Electrical Engineering and Computer Science, Catholic University of America, Washington, DC 20064 USA (e-mail: plaku@cua.edu).

D. Sofge is with the Naval Research Laboratory, Washington, DC 20375 USA (e-mail: donald.sofge@nrl.navy.mil).

Digital Object Identifier 10.1109/TASE.2015.2443033

## I. Introduction

UAVs ARE becoming central in environmental monitoring, search-and-rescue missions, package delivery, target tracking, and many other applications. In addition, with UAVs such as ARDrone and AscTec Pelican quadcopters becoming more commercially available, they are an economically feasible option for deployment in autonomous aerial missions.

Toward increasing the autonomy of UAVs, this paper proposes a reactive motion-planning approach, termed Planner for Autonomous Risk-sensitive (CoveragePARCov), for persistent coverage of risk-sensitive areas by a team of UAVs. By providing persistent coverage, PARCov seeks to reduce the wait time for each region, i.e., elapsed time since the region was last surveyed. This is motivated by many applications where the area being surveyed is often much larger than the area that can be covered by the team at one time. Moreover, in order to utilize the information gathered during surveillance, it becomes important to maintain high sensor data quality. Accounting for risk is also crucial in many aerial missions. In surveillance, the risk could correspond to the likelihood of a quadcopter being detected by a hostile agent on the ground. Such risk can be modeled, for example, as a costmap over a grid representation of the ground level. PARCov can accommodate various risk metrics that decrease in value as the altitude increases. Costmaps have been used in many path-planning approaches [1]–[3] to compute a low-cost path to a given goal. PARCov instead uses the costmaps to enable a team of UAVs to persistently survey an area while reducing risk and maintaining high sensor data quality.

Achieving persistent coverage while reducing risk and maintaining high sensor data quality presents significant challenges. As risk and sensor data quality often increase when flying closer to the ground, it becomes challenging to find an optimal altitude. Scalability is also important to ensure that the approach can be applied to an increasing number of quadcopters.

### A. Related Work

Related work has often focused on individual aspects of coverage, persistency, risk, and sensor data quality. In fact, there is an extensive body of work on coverage methods where a team of robots seeks to map out an environment, ranging from those that assume that the environment is fully known, partially known, to completely unknown [4]–[6]. Often these approaches leverage the idea of guiding the robots toward centers of Voronoi regions in order to maximize coverage. Other approaches partition the environment into polygonal regions and execute predefined motion patterns to cover each region [7]. Dynamic task distribution is proposed in [8]–[10] in order to determine which regions

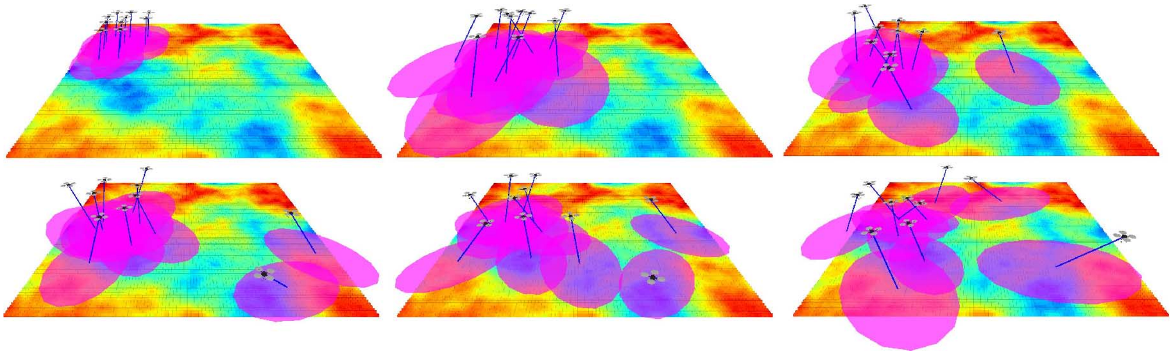


Fig. 1. Snapshots of PARCov at different iterations showing how the quadcopters cover the designated area. The risk model is shown as a heatmap with red indicating high risk and blue indicating low risk. Figures may be best viewed in color and on screen.

should be covered by each AUV in an effort to enhance cooperative coverage and searching. Apker *et al.* [11] propose a bio-inspired coverage approach which models the search space as pastures and the UAVs as grazing animals that seek to consume the available information. Other coverage approaches draw from path planning where the objective is to reach one or more predefined goal regions [12]–[14]. The idea is to decompose the environment into regions, solve a TSP to determine the order in which to visit the regions, and rely on path planning to follow the regions. Path-planning approaches range from discrete search [1], [2], evolutionary algorithms [15]–[19], optimization [20], to sampling-based [21]–[23]. The problem setting in these coverage and path-planning approaches does not account for other objectives, such as reducing risk, maintaining high sensor data quality, or providing persistent coverage.

To provide persistent coverage, Kuhlman *et al.*, [24] developed a closed-loop path optimization for a single UAV which also sought to maximize the information gained from information-rich areas. Communication-range constraints are taken into account in [25] so that UAVs are able to maintain communication while visiting desired regions and avoiding forbidden regions. Cheng *et al.*, [26] plan time-optimal trajectories for a single UAV providing sensor coverage of urban structures. Other approaches assume that the robot travels along predefined paths and focus on adjusting the velocity [27]. Cassandras *et al.*, [28] develop an optimal control approach for persistent monitoring in 1-D mission space. Huynh *et al.*, [29] propose control policies for the persistent-patrol problem in order to reduce the expected time it takes to detect an incident. Cooperative control techniques are developed in [30] to reduce the likelihood of UAVs being detected by radars. Other techniques, based on sequential decision processes, computer vision, dynamic programming, and physiomimetics, have been developed for target tracking [31]–[34].

### B. Contribution

Even though significant progress has been made, it remains challenging to achieve persistent coverage and scalability while reducing risk and maintaining high sensor data quality. While related work has focused on individual aspects of these challenges, PARCov offers an important contribution by combining persistent coverage, risk, and sensor data quality. This is made possible by leveraging simple interactions among UAVs to promote an emergent behavior that enables the quadcopters to maximize coverage. Fig. 1 provides some illustrations of how the quadcopters cover the area. Persistency is achieved by giving priority to regions that have the largest wait times. An optimization formulation is developed in order to compute optimal altitudes for each quadcopter so that it reduces the risk and maintains high sensor data quality. Scalability is obtained by avoiding explicit coordination among UAVs. Instead, each UAV utilizes the information provided by the riskmap and the wait times to determine the next region it should survey. More specifically, PARCov maintains an uncertainty measure designed to keep track of the regions that have been surveyed and the times they were last surveyed. In this way, the uncertainty measure of a region increases the longer the region remains out of the sensor footprint of the quadcopters. PARCov combines the uncertainty measure with the risk into a cost function. In order to reduce the risk, the cost function increases as risk increases. To promote coverage of regions with large wait times, the cost function decreases as uncertainty increases. As a result, PARCov seeks to move each quadcopter toward a low-cost region in its vicinity. Such motions have the desired effect of reducing both the risk and the uncertainty which promotes persistent coverage of risk-sensitive areas. Livelock is avoided since all the UAVs share the same riskmap and wait-time information. This information sharing is the only centralized aspect of PARCov. Although not the focus of this work it is possible to use a peer-to-peer distributed hash-table [35] to store this information. Experimental validation is provided both in simulation and with real AscTec Pelican quadcopters.

A preliminary version of PARCov appeared as a symposium proceeding [36]. This paper offers several improvements over the preliminary version such as the introduction of the uncertainty measure and its combination with the risk into the cost function to effectively guide the quadcopters toward low-cost regions. This paper also provides extended experimental evaluation using different risk models and an increasing number of UAVs. While the preliminary version was limited to simulation, this paper incorporates PID controllers with PARCov and is applied to real AscTec Pelican quadcopters.

## II. Problem Formulation

The problem considered in this paper is to enable a team of quadcopters to persistently cover a given area $\mathcal{A}$ while reducing risk and maintaining high sensor data quality. The models used in this paper for sensor coverage, sensor data quality, and risk are described next.

*a) Area Coverage and Persistency:* Let $Q = \{q_1, \ldots, q_n\}$ denote the set of quadcopters. Each quadcopter $q_i$ has a sensor mounted at a fixed angle $\phi$. The area on the $xy$ plane sensed by $q_i$ at time $t$ is denoted by $\mathrm{SensedArea}_{q_i}(t)$. PARCov can accommodate different sensor footprints. Motivated by recent applications of UAVs to provide illumination [37], the experiments in this paper use spotlight sensors, so the $\mathrm{SensedArea}_{q_i}(t)$ corresponds to an ellipse defined by the position and orientation of $q_i$, the mounting angle $\phi$, and the conic aperture $\alpha$. Fig. 1 provides an illustration. The overall area sensed by the team is then defined as

$$\mathrm{SensedArea}(t) = \bigcup_{q_i \in Q} \mathrm{SensedArea}_{q_i}(t). \tag{1}$$

As the number of quadcopters might not be sufficiently large to cover the entire area at a time $t$, i.e., $\mathrm{SensedArea}(t) \ll \mathrm{area}(\mathcal{A})$, PARCov seeks to reduce the average wait time of regions in $\mathcal{A}$. In this way, rather than remain still, the quadcopters will fly from one region to the next in order to persistently cover $\mathcal{A}$.

*b) Sensor Data Quality:* PARCov also seeks to maintain high sensor data quality, which is needed in many surveillance and target-tracking applications. For modeling, it is assumed that there is an optimal altitude, denoted by $\mu_{\mathrm{sq}}$, which yields the highest sensor data quality. The optimal altitude, which is a user-defined argument, depends on the particular task and can vary from situation to situation. As an example, the optimal altitude when tracking a person can be smaller than when tracking a tank since the tank is larger and faster. Furthermore, it is assumed that there is an exponential decrease in the sensor data quality as the deviation from $\mu_{\mathrm{sq}}$ increases. More specifically, the sensor data quality is modeled as a distribution with mean $\mu_{\mathrm{sq}}$ and standard deviation $\sigma_{\mathrm{sq}}$, i.e.,

$$\mathrm{SQ}(z) = \exp\left(-\left(\frac{z}{\cos\phi} - \mu_{\mathrm{sq}}\right)^2 \Big/ \left(2\sigma_{\mathrm{sq}}^2\right)\right) \tag{2}$$

where $z$ is the altitude of the quadcopter and $\phi$ is the angle at which the sensor is mounted.

*c) Risk Model:* Another objective of PARCov is to minimize the risk associated with the task assigned to the team of the quadcopters. To provide generality, the risk at the ground level is defined first via a function $R_0 : \mathbb{R}^2 \to [0, 1]$. In order to increase robustness and reduce the gap between simulation and reality, PARCov can accommodate noisy estimates of the true risk. In such scenarios, a quadcopter will not have access to $R_0(x, y)$ but instead will know $\tilde{R}_0(x, y)$ where

$$\tilde{R}_0(x, y) = R_0(x, y) + \mathrm{noise}(x, y). \tag{3}$$

PARCov can incorporate different noise models. In the experiments, the noise is generated uniformly at random.

As the altitude increases, it is assumed that the risk decreases exponentially. In this way, $R(x, y, z)$ which represents the risk at the location $(x, y, z) \in \mathbb{R}^3$, is defined as

$$R(x, y, z) = \tilde{R}_0(x, y) \cdot \exp\left(-\frac{z^2}{K \cdot \tilde{R}_0(x, y)^2}\right) \tag{4}$$

where $K$ is a scaling constant. As a result, quadcopters need to fly at higher altitudes over areas with high ground-level risk in order to reduce the likelihood of being detected. Note that this model does not account for the risk posed by the wind, other environmental disturbances, adversarial UAVs which can become present at any altitude. The focus of this work is on risk which is monotonic with respect to the altitude.

PARCov can work with any ground-risk model. The ground-risk models used in the experiments are generated by the diamond-square algorithm [38], which constructs random heatmaps that have values from zero to one. Random heatmaps have also been used in [39] to represent random risk for path-planning problems. Figs. 1 and 3 provide several illustrations of different heatmaps generated by the diamond-square algorithm. More details are provided in Section IV.

*d) Problem Statement:* The problem considered in this paper can now be stated as follows: Given an area $\mathcal{A}$ to be surveyed, risk and sensor data quality models, and an initial placement of the quadcopters, move the quadcopters so that they persistently cover $\mathcal{A}$ while reducing the risk and maintaining high sensor data quality.

PARCov assumes that each quadcopter knows its position at all times as well as the risk value [which could be noisy, as described in (3)]. In addition, the quadcopters share the wait-time information so that all quadcopters know the time when each region was last surveyed. When going from simulation to physical experiments, it is assumed that the control profiles of the quadcopters can be computed in real time. This is indeed supported by the experiments with the AscTec Pelican quadcopters.

## III. Method

Planning in PARCov occurs in two stages. During the first stage, PARCov plans motions of the quadcopters in $xy$ to promote persistent area coverage. During the second stage, PARCov adjusts the altitude of the quadcopters to minimize the risk while maximizing sensor data quality. Pseudocode is given in Algorithm 1. Descriptions of the main steps follow.

### *A. Planning Motions in 2D*

In order to promote area coverage and persistency while reducing the risk, PARCov relies on a cost function $\mathrm{COST} : \mathbb{R}^2 \times \mathbb{N} \to \mathbb{R}^+$ to determine the $xy$ motions of each quadcopter, where $\mathrm{COST}(x, y, t)$ denotes the cost associated with the point $(x, y)$ at time $t$ (expressed as an iteration count). Note that the cost function changes dynamically based on the movements of the quadcopters. In other words, at time $t$, the current cost values are known but not the cost values at later times. The cost function is constructed by combining the ground-level risk $\tilde{R}_0(x, y)$ with an uncertainty measure $U : \mathbb{R}^2 \times \mathbb{N} \to \mathbb{R}^+$, where $U(x, y, t)$ indicates the time that has elapsed since $(x, y)$ was last sensed by a quadcopter. In this way, the uncertainty associated with $(x, y)$ increases as $(x, y)$ goes longer and longer without being sensed. This paper allows for any combination of $\tilde{R}_0$ and $U$. Generally

$$\mathrm{COST}(x, y, t) \overset{\mathrm{def}}{=} \mathrm{COMBINE}(\tilde{R}_0(x, y), U(x, y, t), t) \tag{5}$$

where COMBINE is a user-defined function. For the experiments, the cost is defined as

$$\mathrm{COST}(x, y, t) = \alpha\tilde{R}_0(x, y) + \exp(t - U(x, y, t)) \tag{6}$$

where $\alpha$ is a scaling constant. In this way, to reduce the detection risk, the cost increases as risk increases. To promote coverage of areas with high uncertainty, the cost, decreases exponentially as the uncertainty increases.

**Algorithm 1: Pseudocode for PARCov**

```
input:
  • area to be surveyed $\mathcal{A}$
  • quadcopters $Q = \{q_1, \ldots, q_n\}$
  • sensor model where $\text{SensedArea}(q_i) \subseteq \mathcal{A}$
  • sensor data quality model $SQ : \mathbb{R} \to [0,1]$
  • noisy ground-level risk $\tilde{R}_0 : \mathbb{R}^2 \to [0,1]$
  • risk model $R : \mathbb{R}^3 \to [0,1]$
  • COMBINE: combines $\tilde{R}_0$ with the uncertainty measure $U$
output: new positions and orientations of each quadcopter
 1: $U_{\text{grid}} \leftarrow$ INITUNCERTAINTYGRID(); $t \leftarrow 1$
 2: while FINISHED() = false do
 3:   for $q \in Q$ do
 4:     $(v', \beta') \leftarrow \text{DIRANDANGLE}_{\langle \tilde{R}_0, U_{\text{grid}}, \text{COMBINE} \rangle}(q, t)$
 5:     $\begin{bmatrix} x' \\ y' \end{bmatrix} \leftarrow \begin{bmatrix} q.x \\ q.y \end{bmatrix} + \text{step} \cdot \frac{v'}{\|v'\|}$
 6:     $z' \leftarrow \text{DETERMINEALTITUDE}_{\langle R, SQ \rangle}(x', y')$
 7:     SETTARGETPOSANDORIENTATION$(q, x', y', z', \beta')$
 8:     CONTROLLER$(q, x', y', z', \beta')$
 9:     UPDATEUNCERTAINTYGRID$(U_{\text{grid}}, q, t)$
10:   $t \leftarrow t + 1$
```

**Algorithm 2: $\text{DIRANDANGLE}_{\langle \tilde{R}_0, U_{\text{grid}}, \text{COMBINE} \rangle}(q, t)$**

```
define COST$(x, y, t)$ = COMBINE$(\tilde{R}_0(x, y), U(x, y, t), t)$
 1: MINAVGCOST $\leftarrow \infty$; $v' \leftarrow (0, 0)$; $\beta' \leftarrow 0$
 2: orientations $\leftarrow$ GETRANDOMSAMPLES$(0, 2\pi)$
 3: for $\beta \in$ orientations do
 4:   $\xi \leftarrow$ ENLARGE(SENSEDAREA$(q.x, q.y, q.z, \beta), \epsilon)$
 5:   segments $\leftarrow$ GETSEGMENTS$(\xi)$
 6:   for $s \in$ segments do
 7:     points $\leftarrow$ GETPOINTS$(s)$
 8:     avgCost $\leftarrow \frac{1}{|\text{points}|} \sum_{(x,y) \in \text{points}} \text{COST}(x, y, t)$
 9:     if minAvgCost < avgCost then
10:       minAvgCost $\leftarrow$ avgCost
11:       $\beta' \leftarrow \beta$
12:       sumWeight $\leftarrow \sum_{(x,y) \in \text{points}} 1/\text{COST}(x, y, t)$
13:       $v' \leftarrow \begin{bmatrix} -q.x \\ -q.y \end{bmatrix} + \sum_{(x,y) \in \text{points}} \frac{1/\text{COST}(x,y,t)}{\text{sumWeight}} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$
14:   return $(v', \beta')$
```

As an implementation note, the uncertainty measure is maintained by first imposing a grid $U_{\text{grid}}$ over the $xy$ bounding box of the area $\mathcal{A}$ being surveyed. The uncertainty grid is used to keep track of the regions in $\mathcal{A}$ that have been surveyed and the times that they were last surveyed. More specifically, $U_{\text{grid}}(c)$ returns the time, as an iteration count, when the grid cell $c$ was last sensed by a quadcopter. The uncertainty grid is initialized by setting each $U_{\text{grid}}(c)$ to zero (Algorithm 1:1). Whenever a grid cell $c$ is sensed by some quadcopter, i.e., $c \subseteq \text{SensedArea}_{q_i}(t)$, the current iteration count $t$ is stored in $c$ (Algorithm 1:9). The uncertainty measure $U(x, y, t)$, which keeps track of the time that has elapsed since $(x, y)$ was last sensed, is then computed as $t - U_{\text{grid}}(c)$ where $t$ is the current iteration count and $c$ is the grid cell that contains $(x, y)$. Note that $c$ can be computed in constant time from $(x, y)$.

Illustrations of the uncertainty measure and the cost function at various time instances are provided in Fig. 2. As the algorithm progresses the uncertainty of the area changes and so does the associated cost. These measures are vital in order to effectively plan the motions of the quadcopters.

To reduce the uncertainty and detection risk, PARCov seeks to move each quadcopter toward a low-cost region in its vicinity. Pseudocode is given in Algorithm 2. PARCov first generates a set of candidate orientations $\{\beta_1, \ldots, \beta_\ell\}$ at random, i.e., $\beta_i = \text{RAND}(0, 2\pi)$ (Algorithm 2:2). Let $\text{SensedArea}(q, \beta)$ denote the area that would be sensed by $q$ when keeping the same position but setting the orientation to $\beta$. As an illustration, the sensed area for a spotlight sensor model would be an ellipse defined with respect to the parameter $\omega \in [0, 2\pi]$ as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \times \begin{bmatrix} z \cdot \tan(\phi - \alpha) + A_M \cdot (1 + \cos\omega) \\ A_m \cdot \sin\omega \end{bmatrix} \quad (7)$$

where $(x, y, z)$ is the position of the quadcopter, $\phi$ is the angle at which the sensor is mounted, $\alpha$ is the conic aperture, $A_M = z \cdot \tan(\phi + \alpha) - z \cdot \tan\phi$ and $A_m = (z \cdot \tan\alpha)/(\cos\phi)$ are the major and minor axis, respectively.

To take advantage of locality, PARCov considers $\xi(q, \beta)$ which represents $\text{SensedArea}(q, \beta)$ enlarged by some $\epsilon > 0$ (Algorithm 2:4). To determine where to move $q$ so that it can lower the cost, PARCov generates several segments along the perimeter of $\xi(q, \beta)$. Specifically, let $p_i$ denote the point on the perimeter of $\xi(q, \beta)$ corresponding to the angle parameter $i \cdot \zeta$, where $\zeta$ defines the sampling value (set to 10° in the experiments). The $i$th segment is then obtained by connecting $p_i$ to $p_{i+1}$. A cost is computed for each segment $s$, denoted by $\text{SegmentCost}(s)$, as the average of the costs of a number of equally spaced points along $s$. PARCov will then move the quadcopter $q$ toward the segment with the minimum cost, i.e.,

$$s = \underset{s' \in \text{AllSegments}}{\arg\min} \ \text{SegmentCost}(s') \quad (8)$$

where $\text{AllSegments} = \bigcup_{\beta \in \text{orientations}} \text{segments}(\xi(q, \beta))$.

The new orientation of $q$ is set to $\beta \in \{\beta_1, \ldots, \beta_\ell\}$ from which $\xi(q, \beta)$ that contains the segment $s$ with minimum cost was derived (Algorithm 2:11). The new direction is set by taking a weighted average of the points along $s$ (Algorithm 2:12–13), i.e.,

$$\begin{bmatrix} -q.x \\ -q.y \end{bmatrix} + \sum_{(x,y) \in \text{points}(s)} \frac{1/\text{COST}(x, y, t)}{w} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (9)$$

where $w = \sum_{(x',y') \in \text{points}(s)} 1/\text{COST}(x', y', t)$. In this way, points in $s$ associated with lower costs exert a higher influence when determining the new direction. Finally, the $xy$

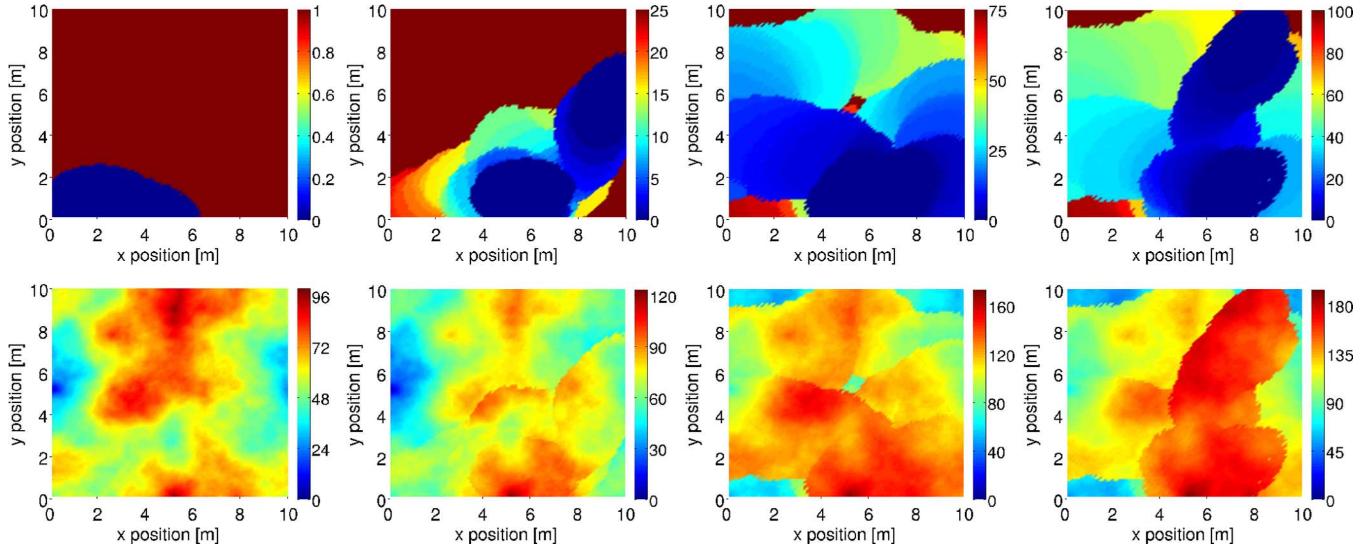


Fig. 2. Instances of the uncertainty measure $U$ (top row) and cost function (bottom row) after 1, 25, 75, 100 iterations of PARCov. For the uncertainty measure, the color in the spectrum represents the time that has elapsed, as an iteration count, since the point $(x, y)$ was last visited. As blue represents recently visited areas, the uncertainty increases from blue to red, as indicated in the adjacent color spectrum scale. Recall that the cost function (bottom row) is computed by combining the uncertainty measure with the ground-level risk (which is shown in Fig. 1). Figures may be best viewed in color and on screen.

position is set by taking a small step along the new direction (Algorithm 1:5).

Such planning has desirable emergent properties for the team. By sharing the same cost function quadcopters will act cooperatively to cover the unvisited areas without having to explicitly coordinate with one another. As each quadcopter moves towards a low-cost area in its vicinity, it becomes less likely for the quadcopters to cluster together. In fact, suppose that several quadcopters are moving towards the same segment $s$. As soon as $s$ is sensed by a quadcopter, the uncertainty associated with $s$ becomes zero so its cost increases. As a result, the other quadcopters will move away from $s$ toward other segments that have lower costs.

Moreover, quadcopters have the flexibility to leave and join the team at any time. Since PARCov does not explicitly divide the area among the quadcopters participating in the task, if a quadcopter leaves the area, it would simply no longer update the uncertainty measure. The rest of the team would have no knowledge that it left so they would continue to update the uncertainty measure and move toward areas with low cost. Similarly, a new quadcopter can join the team at any time by accessing the cost function and updating the uncertainty measure. The flexibility of leaving and joining the team as needed is particularly important for missions that combine persistent coverage and target tracking. When the team covering the area detects moving targets, a number of quadcopters from the team can be deployed to track them while the rest continue to survey the area.

### B. *Determining the Altitude*

After computing the $xy$ position, the new target altitude is determined by optimizing an objective function that maximizes the sensor data quality and minimizes the risk, i.e.,

$$H(x, y, z) = \mathrm{SQ}(z) - R(x, y, z). \tag{10}$$

In this way, the optimal altitude corresponds to the $z$ value that maximizes $J$ for a given $x, y$, i.e.,

$$\mathrm{DetermineAltitude}(x, y) \overset{\mathrm{def}}{=} \underset{z \in [z_{\min}, z_{\max}]}{\arg\max} \; H(x, y, z). \tag{11}$$

Note that $H(x, y, z)$ is nonlinear since $\mathrm{SQ}(z)$ and $R(x, y, z)$ involve exponential, polynomial, and trigonometric terms. Nonlinear optimization solvers, such as SciPy, can then be used to numerically compute the altitude that maximizes $H$.

### C. *Controller and Updates to the Uncertainty Grid*

A PID controller is employed to steer each quadcopter toward the new position, altitude, and orientation (Algorithm 1:7–8). The PID controller used in simulation is the same as the one we use to steer the real AscTec Pelican quadcopters toward desired targets. After each motion step, the uncertainty grid $U_{\mathrm{grid}}$ is updated accordingly to account for the newly sensed area (Algorithm 1:7). Since the quadcopters share $U_{\mathrm{grid}}$, the team will dynamically react to new information. In particular, when a quadcopter decreases its altitude there will be more uncovered space around it so other quadcopters will move in to cover these areas. These dynamic adjustments, as shown next, make it possible to efficiently cover the area while maintaining high sensor data quality and reducing the risk.

## IV. Experiments and Results

The performance of PARCov was tested in simulation using an increasing number of quadcopters and various risk maps. Experiments also tested the robustness of PARCov with respect to risk and control noise. Since no other approach is available for the problem setting considered in this paper, an algorithm that moves the quadcopters at random was used to provide a baseline comparison. Experiments were also conducted with real AscTec Pelicans.

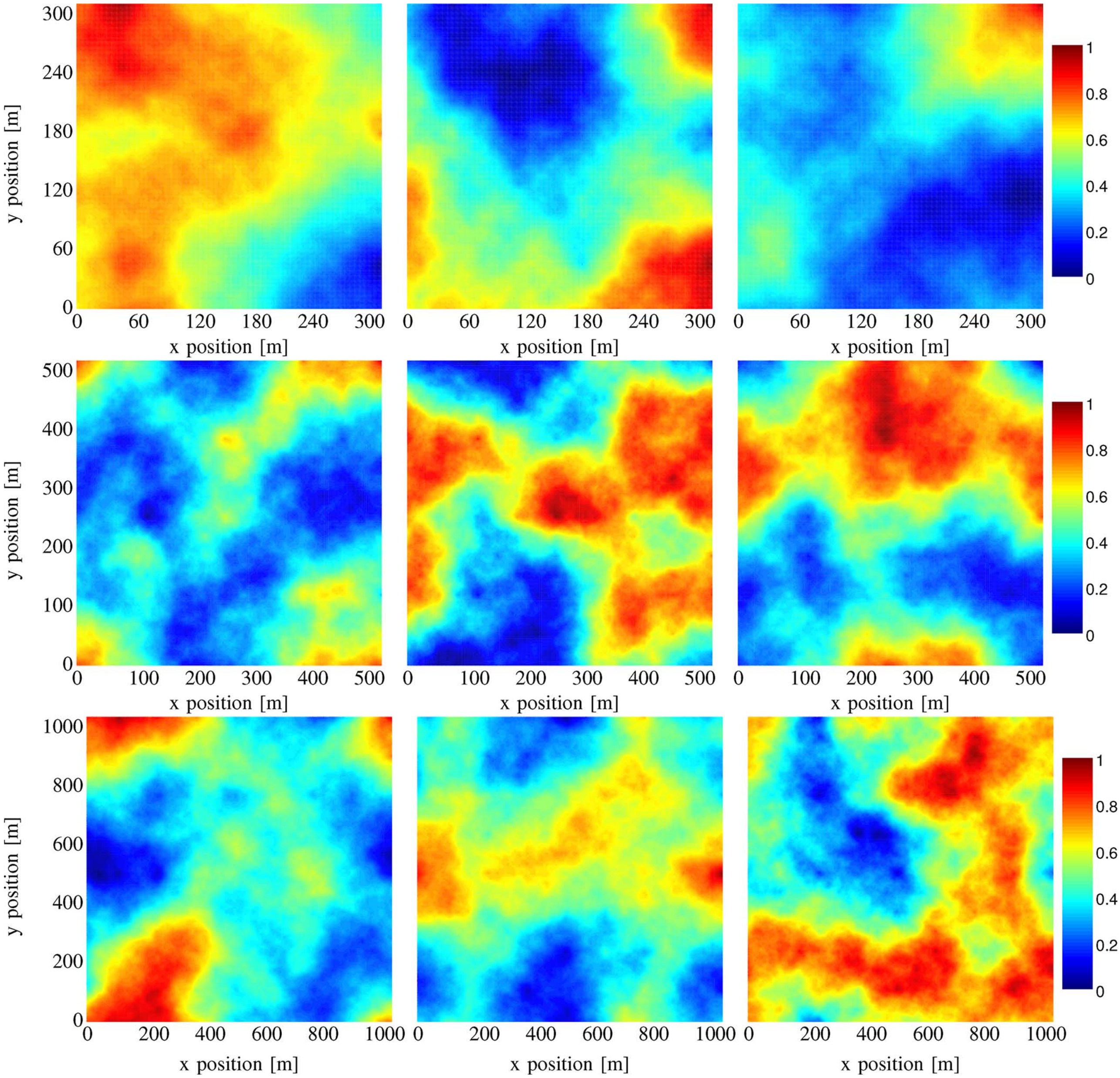


Fig. 3. Ground-level risk heatmaps used in the experiments where the color indicates the risk value as shown in the color spectrum. (top row) Scenes 1, 2, 3; size: 300 m × 300 m. (middle row) Scenes 4, 5, 6; size: 500 m × 500 m. (bottom row) Scenes 7, 8, 9; size: 1000 m × 1000 m.

### A. Scenes

Experiments were conducted with three different scene sizes: 300 m × 300 m, 500 m × 500 m, and 1000 m × 1000 m. In each case, the quadcopters were required to fly between $z_{\min} = 5$ m and $z_{\max} = 400$ m. For each scene size, three different heatmaps, representing different ground-level risks, were generated by using the diamond-square algorithm [38]. Fig. 3 shows all the nine scenes used in the experiments.

Table I shows the sensor footprint of a single quadcopter when placed at the height that maximizes the sensor data quality when there is no risk. The sensor footprint is shown as a percentage over the total area being surveyed. The table indicates that a large number of quadcopters would be needed in order to cover the scene by standing still. This motivates the need for an approach that can effectively obtain persistent coverage in order to survey the areas that are outside the current sensor footprint.

TABLE I
INDIVIDUAL SENSOR FOOTPRINT AT THE HEIGHT THAT MAXIMIZES SENSOR DATA QUALITY AS A PERCENTAGE OF THE TOTAL AREA

| $300m \times 300m$ | $500m \times 500m$ | $1000m \times 1000m$ |
|---|---|---|
| 0.81% | 0.29% | 0.07% |

### B. Performance Criteria

The performance of PARCov is measured according to several criteria: current coverage $(C_1)$, cumulative coverage $(C_2)$, 90%-persistent coverage $(C_3)$, sensor quality $(C_4)$, risk $(C_5)$, and average wait time $(C_6)$, as described below next.

Current coverage $(C_1)$ measures the mean percentage of the area covered at each iteration of PARCov, i.e.,

$$C_1(t) = 100\,\frac{1}{t}\sum_{i=1}^{t}\frac{\text{SensedArea}(i)}{\text{area}(\mathcal{A})} \tag{12}$$

where $t$ denotes the current iteration count and, as defined in Section II, $\text{SensedArea}(i) = \text{SensedArea}_{q_1}(i) \cup \ldots \cup \text{SensedArea}_{q_n}(i)$ denotes the area sensed by the quadcopters at the $i$th iteration. In the experiments, after updating the uncertainty grid $U_{\text{grid}}$ (Algorithm 1:9) for all the quadcopters, $\text{SensedArea}(i)$ is computed by adding the areas of the grid cells in $U_{\text{grid}}$ that are sensed during the $i$th iteration. Current coverage $(C_1)$ provides a measure of how well the quadcopters spread out to cover the area at any given instance.

Cumulative coverage $(C_2)$ measures the percentage of the area covered over all the iterations of PARCov, i.e.,

$$C_2(t) = 100 \frac{1}{\text{area}(\mathcal{A})} \bigcup_{i=1}^{t} \text{SensedArea}(i). \tag{13}$$

It is expected, as the iteration count $t$ increases, PARCov will cover the entire area. In the experiments, $C_2(t)$ is computed by adding up the areas of all the grid cells in $U_{\text{grid}}$ that have an iteration count greater than zero.

The third criteria, referred to as 90%-persistent coverage $(C_3)$, is designed to measure coverage persistency. In particular, it measures the average number of iterations to repeatedly reach 90% cumulative coverage. More specifically, we keep a counter to track the number of iterations to reach 90% cumulative coverage. After reaching it, the value of the counter is pushed onto a list and the cumulative coverage and the counter are set back to zero. At time $t$, the measure $C_3(t)$ is then computed as the average value in the list of counters. As an implementation note, the cumulative coverage needed for the purposes of computing $C_3(t)$ is obtained as

$$100 \frac{1}{\text{area}(\mathcal{A})} \bigcup_{i=1+t_{\text{last90}}}^{t} \text{SensedArea}(i) \tag{14}$$

where $t_{\text{last90}}$ is the iteration count when 90% cumulative coverage was reached. This is computed by adding up the areas of all the grid cells in $U_{\text{grid}}$ that have an iteration count greater than $t_{\text{last90}}$. In this way, the 90%-persistent coverage criterion measures how long it takes the quadcopters to cover 90% of the space starting from many different configurations.

Sensor quality $(C_4)$ is measured by considering the sensor-quality model SQ (Section II) and the distances from the quadcopters to the grid cells that they cover. More specifically, consider a grid cell $c \in U_{\text{grid}}$. The sensor quality associated with $c$ at iteration $i$ is defined by taking the maximum sensor quality among the quadcopters that sense $c$, i.e.,

$$\text{SQ}(c, i) = \max_{\substack{q \in Q \text{ and} \\ c \subseteq \text{SensedArea}_q(i)}} \text{SQ}(\text{dist}(c, q)) \tag{15}$$

where $\text{dist}(c, q)$ denotes the distance from the center of the cell $c$ to the $(q.x, q.y, q.z)$ position of the quadcopter $q$. The sensor quality associated with $U_{\text{grid}}$ at iteration count $i$ is defined as the average of the sensor qualities associated with the grid cells sensed at the $i$th iteration, i.e.,

$$\text{SQ}(U_{\text{grid}}, i) = \sum_{c \in \text{SensedCells}(i)} \frac{\text{SQ}(c, i)}{|\text{SensedCells}(i)|} \tag{16}$$

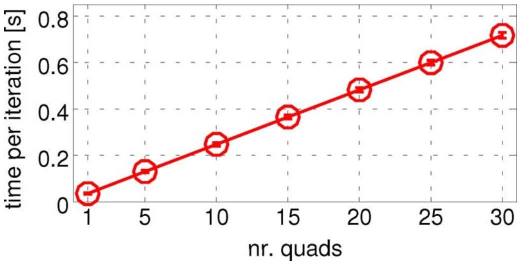


Fig. 4. Trajectories taken by two quadcopters for two different scenes. The ground-level risk $R_0$ is shown as a heatmap.

where $\text{SensedCells}(i) = \{c : c \in \text{cells}(U_{\text{grid}}) \text{ and } c \subseteq \text{SensedArea}(i)\}$. Then, the overall sensor-quality measure $(C_4)$ is defined as the average sensor-quality of $U_{\text{grid}}$ over the iterations $1, \ldots, t$ scaled to a percentage, i.e.,

$$C_4(t) = 100 \frac{1}{t} \sum_{i=1}^{t} \text{SQ}(U_{\text{grid}}, i). \tag{17}$$

The risk $(C_5)$ measures the average detection risk over all the quadcopters and over the iterations $1, \ldots, t$ scaled to a percentage, i.e.,

$$C_5(t) = 100 \frac{1}{t} \sum_{i=1}^{t} \sum_{q \in Q} \frac{R(q.x, q.y, q.z)}{|Q|}. \tag{18}$$

The average wait time $(C_6)$ is used to show that no part of the area being surveyed remains unsensed for a long time. More specifically, at the end of each iteration $i$ (Algorithm 1:10), the average wait time for a grid cell is computed as

$$\text{WaitTime}(i) = \sum_{c \in \text{cells}(U_{\text{grid}})} \frac{i - U_{\text{grid}}(c)}{|\text{cells}(U_{\text{grid}})|}. \tag{19}$$

These wait times are stored in a list and $C_6(t)$ is computed by taking their average, i.e.,

$$C_6(t) = \frac{1}{t} \sum_{i=1}^{t} \text{WaitTime}(i). \tag{20}$$

### C. Results

Before presenting quantitative results, we provide some qualitative illustrations to show PARCov in action. Fig. 1 shows how the quadcopters cover the designated area. Using the cost function obtained by combining the uncertainty measure with the risk, the quadcopters start moving toward areas associated with low cost.

Another illustration of PARCov in action is provided in Fig. 4, which shows trajectories taken by two quadcopters. Note how the quadcopters increase their altitude when surveying areas designated as high risk and reduce their altitude when going over low-risk areas.

Fig. 5 shows the runtime per iteration, where an iteration ends when the new position and orientation is determined for each quadcopter (Algorithm 1:3–9). The results show that the runtime per iteration increases linearly with the number of quadcopters. These results, as all the experiments, were obtained on an Intel Core i7 machine (CPU: 2.40 GHz, RAM: 8 GB) using

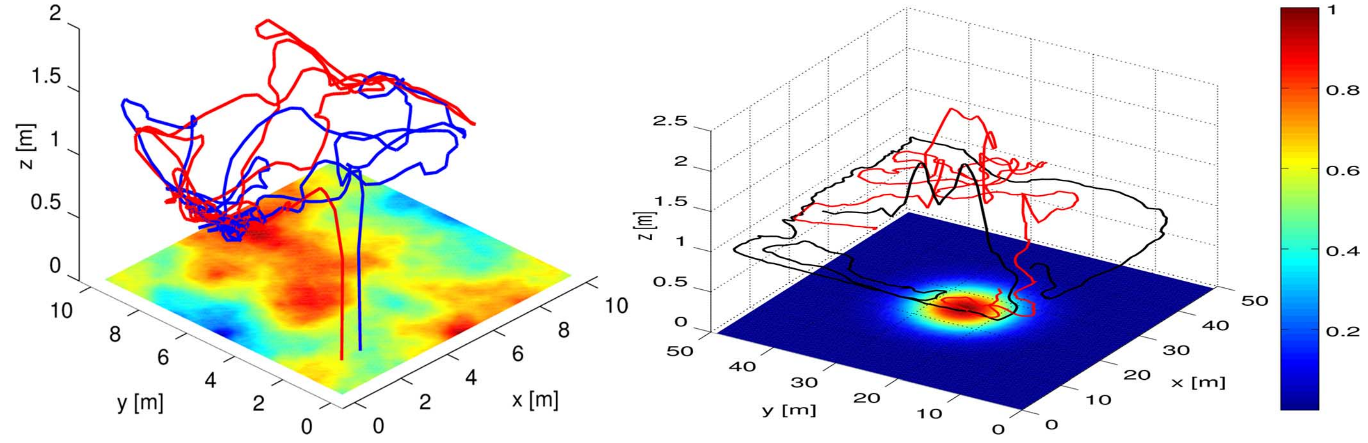


Fig. 5. Runtime per iteration. Bars indicate standard deviation. Results are shown for scene 8 (1000 m × 1000 m) found in Fig. 3.

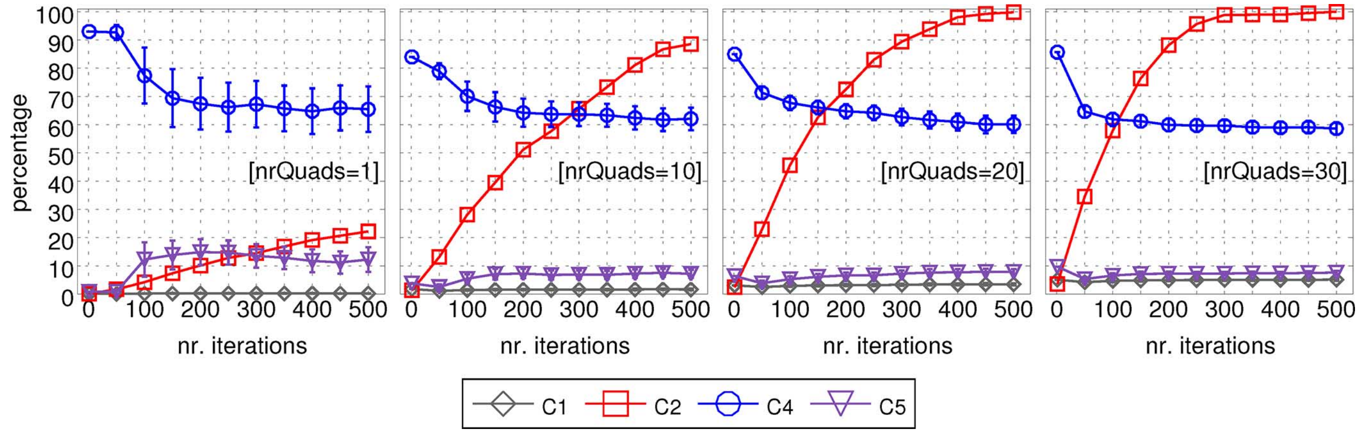


Fig. 6. Performance criteria: current coverage $C_1(t)$, cumulative coverage $C_2(t)$, sensor quality $C_4(t)$, and risk $C_5(t)$ with respect to the number of iterations $t$. The risk noise (3) was drawn uniformly at random with the maximum value set at 0.05. Results are shown for scene 8 (1000 m × 1000 m) found in Fig. 3.

Ubuntu 14.04. Code was written in Python 2.7. ROS rviz was used for visualization.

The rest of the section presents results on the performance of PARCov according to various criteria $(C_1 - C_6)$. Results are also presented that show the performance as a function of number of iterations, number of quadcopters, scene size, risk noise, and control noise, as well as comparisons with a baseline random algorithm. The section concludes with results from experiments with real AscTec Pelican quadcopters.

*1) Results on Various Performance Criteria:* Fig. 6 shows the performance of PARCov in terms of the current coverage $(C_1)$, cumulative coverage $(C_2)$, sensor quality $(C_4)$, and risk $(C_5)$. The results indicate that PARCov effectively guides the quadcopters to quickly cover the designated area. In fact, the cumulative coverage $(C_2)$ increases rapidly with the number of iterations. The current coverage $(C_1)$ also increases with the number of quadcopters. The initial increase with the number of iterations results from the spreading of the quadcopters from the initial configuration which has them all start near each other in a corner of the scene. The figure also shows that PARCov enables the quadcopters to maintain high sensor data quality $(C_4)$ while significantly reducing the risk $(C_5)$.

Fig. 7 shows the performance of PARCov when varying the scene size, risk model, and the number of quadcopters. The algorithm works well for a variety of scenes and ground-risk models. As shown, PARCov effectively surveys the area while maintaining high sensor quality and minimizing risk.

Fig. 8 shows the performance of PARCov in terms of the 90%-persistent coverage criterion $(C_3)$. The results show that the average number of iterations to reach 90% coverage decreases rapidly as more and more quadcopters work together. This quick convergence combined with the data from Fig. 6 shows that PARCov is able to maintain persistent coverage of the area while maintaining low risk and high sensor quality.

Fig. 9 shows the average wait time $(C_6)$ as a function of the number of quadcopters for different scenes. As the number of quadcopters increases, the average wait time decreases since the quadcopters spread through the space cooperatively and therefore cover the space more quickly. These results provide further evidence about the ability of PARCov to maintain persistent coverage.

*2) Baseline Comparison:* Table II provides a summary when comparing PARCov to a random planner which moves the quadcopters in random directions. The random planner serves as a baseline since no other approach is available for the problem setting considered in this paper. Results show that the random planner can obtain cumulative coverage but has difficulty minimizing risk. The random planner also has difficulty obtaining persistent coverage and reducing wait times.

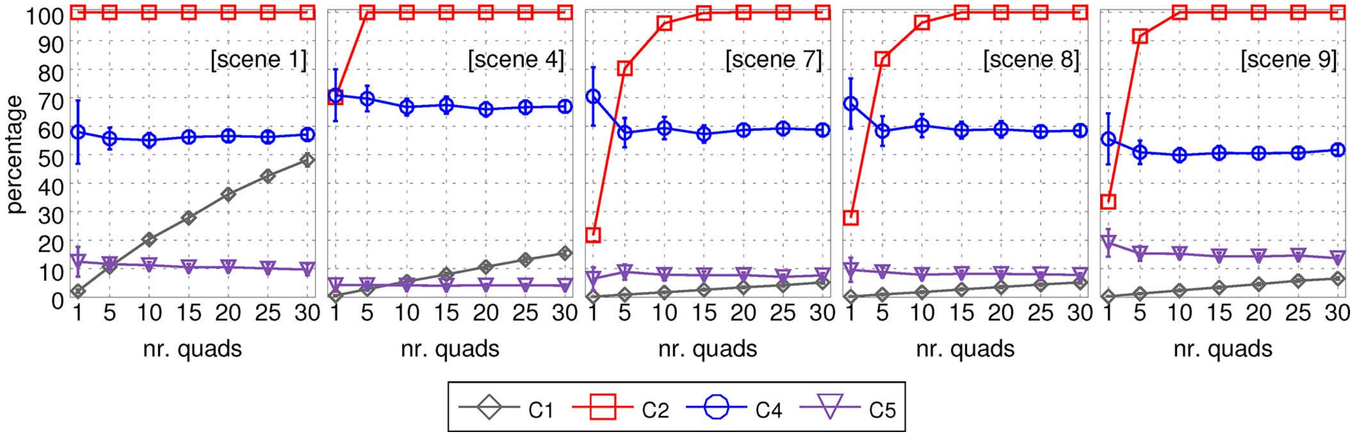


Fig. 7. Performance criteria: current coverage $C_1(750)$, cumulative coverage $C_2(750)$, sensor quality $C_4(750)$, and risk $C_5(750)$ when varying the scene size, the risk model, and the number of quadcopters. The risk noise (3) was drawn uniformly at random from with the maximum value set at 0.05.

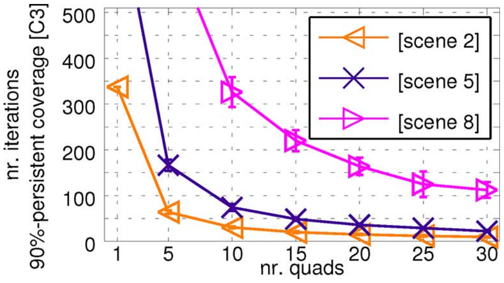


Fig. 8. 90% persistent coverage, i.e., $C_3(750)$, when varying the number of quadcopters and scene size. The risk noise (3) was drawn uniformly at random from with the maximum value set at 0.05.

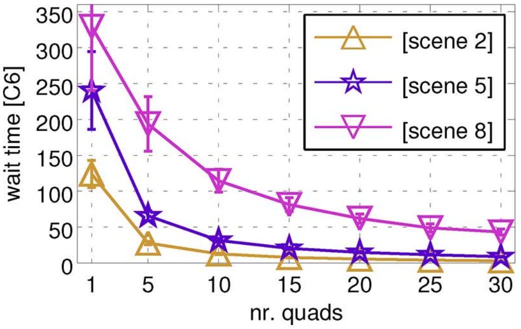


Fig. 9. Mean wait time, i.e., $C_6(750)$, when varying the number of quadcopters and scene size. The risk noise (3) was drawn uniformly at random from with the maximum value set at 0.05.

TABLE II
BASELINE COMPARISON WITH RANDOM PLANNER

| [scene 6] | Random Planner | | PARCov | |
|---|---|---|---|---|
| nr. quads | 10 | 30 | 10 | 30 |
| $C_1(750)$ current coverage | 1.7 | 5.3 | 7.6 | 21.3 |
| $C_2(750)$ cumulative coverage | 91.6 | 99.7 | 100.0 | 100.0 |
| $C_3(750)$ 90%-persistent coverage | 699.0 | 205.3 | 74.2 | 22.3 |
| $C_4(750)$ sensor quality | 88.9 | 84.2 | 56.3 | 55.3 |
| $C_5(750)$ risk | 32.1 | 33.8 | 11.2 | 10.7 |
| $C_6(750)$ wait time | 146.9 | 67.6 | 31.6 | 9.0 |

*3) Performance When Varying Risk Noise:* Table III provides a summary of the results when varying the noise added to the ground-risk level (3). The results show the robustness of PARCov to accommodate different noise levels.

TABLE III
PERFORMANCE CRITERIA WHEN VARYING THE RISK NOISE

| [scene 4] | nr. quads = 10 | | | | nr. quads = 30 | | | |
|---|---|---|---|---|---|---|---|---|
| noise | 0.05 | 0.15 | 0.2 | 0.25 | 0.05 | 0.15 | 0.2 | 0.25 |
| $C_1(750)$ | 5.6 | 5.5 | 5.5 | 5.6 | 15.5 | 15.4 | 15.5 | 15.6 |
| $C_2(750)$ | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| $C_3(750)$ | 89.5 | 94.4 | 104.6 | 96.6 | 28.5 | 29.5 | 28.6 | 28.5 |
| $C_4(750)$ | 66.3 | 68.0 | 67.9 | 66.5 | 66.2 | 67.2 | 66.1 | 67.1 |
| $C_5(750)$ | 4.2 | 4.1 | 4.0 | 4.3 | 4.2 | 4.0 | 4.1 | 4.1 |
| $C_6(750)$ | 37.4 | 40.5 | 40.3 | 40.7 | 11.9 | 12.3 | 12.0 | 12.1 |

TABLE IV
PERFORMANCE CRITERIA WHEN VARYING THE CONTROL NOISE

| [scene 5] | nr. quads = 10 | | | | nr. quads = 30 | | | |
|---|---|---|---|---|---|---|---|---|
| noise | 0 | 0.2 | 0.4 | 0.6 | 0 | 0.2 | 0.4 | 0.6 |
| $C_1(750)$ | 7.5 | 7.7 | 7.6 | 7.5 | 21.8 | 21.3 | 21.8 | 21.4 |
| $C_2(750)$ | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| $C_3(750)$ | 92.0 | 76.8 | 76.9 | 82.9 | 22.1 | 23.2 | 22.5 | 22.6 |
| $C_4(750)$ | 52.8 | 53.8 | 56.1 | 54.5 | 53.9 | 54.0 | 53.3 | 53.9 |
| $C_5(750)$ | 11.3 | 11.5 | 11.5 | 11.0 | 11.8 | 11.4 | 11.7 | 11.3 |
| $C_6(750)$ | 35.7 | 33.1 | 31.9 | 34.7 | 8.8 | 9.2 | 8.8 | 9.0 |

*4) Performance When Varying Control Noise:* In order to determine whether PARCov would be able to withstand an acceptable amount of error in the controller, we have conducted experiments to test the performance by changing the amount of noise that is present in the simulated PID controller. We did this by adding Gaussian noise to the control output of the PID controller. The noise was quantified by altering the standard deviation of the Gaussian function. For the experiments, the mean for the Gaussian was set to zero and the standard deviation was set from 0 to 0.6 meters. This range of values seemed to be an accurate representation of the control noise present on the AscTec Pelicans we used for the practical experimentation. Table IV provides a summary of the results. Since PARCov is purely reactive, we can see that within an acceptable amount of noise PARCov is still able to perform well according to all the metrics we have used.

### *D. Physical Experiments*

We tested PARCov at the Laboratory for Autonomous Systems Research located at the Naval Research Laboratory, Washington, DC, USA. We used two AscTec Pelican quadcopters operated in a 5.2 m × 5.2 m area surrounded by ten Vicon mo-

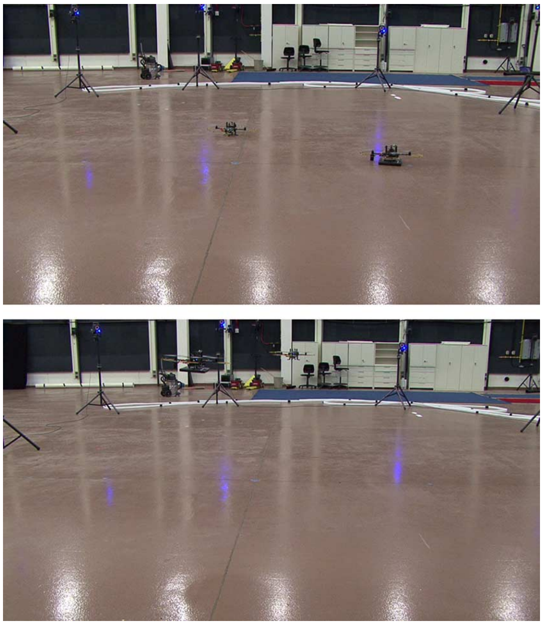


Fig. 10. Experimental setup with two AscTec Pelican quadcopters surrounded by ten Vicon motion capture cameras.

tion capture cameras, as shown in Fig. 10. The quadcopters were equipped with passive motion capture markers which were seen by the Vicon cameras. The Vicon cameras have LEDs which were used to illuminate the passive markers. The 3D positions and the orientations of the quadcopters were sent to a laptop which uses the Robotic Operating System (ROS) and a swarm middleware called ZeroMQ-ROS to compute the control commands and to send them to the quadcopters. ROS is a open-source middleware combined with software libraries and tools to build robotic applications. ZeroMQ-ROS is a middleware for controlling a swarm using a ROS multimaster architecture. The control commands were sent over WiFi through a socket to each of the quadcopters.

Fig. 11 plots the trajectories followed by the AstTec Pelican quadcopters. Fig. 12 shows the performance metrics. The results indicate that PARCov enabled the AscTec Pelican quadcopters to effectively survey the entire area. Note that the quadcopters start at a high-risk area and remain there during takeoff, which takes some time. After that, PARCov guides the quadcopters toward areas the reduce the risk while maintaining high sensor data quality.

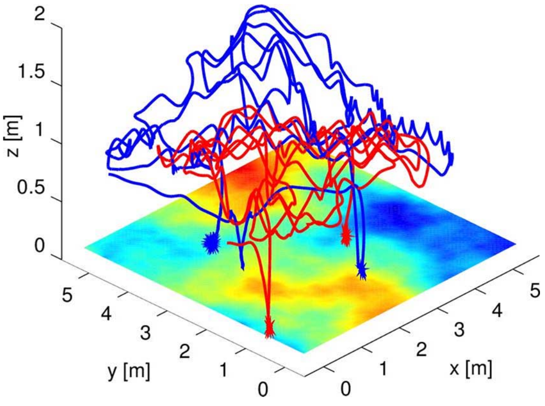


Fig. 11. Trajectories of the two AscTec Pelicans when surveying the area. The oscillations shown are caused by the control noise.

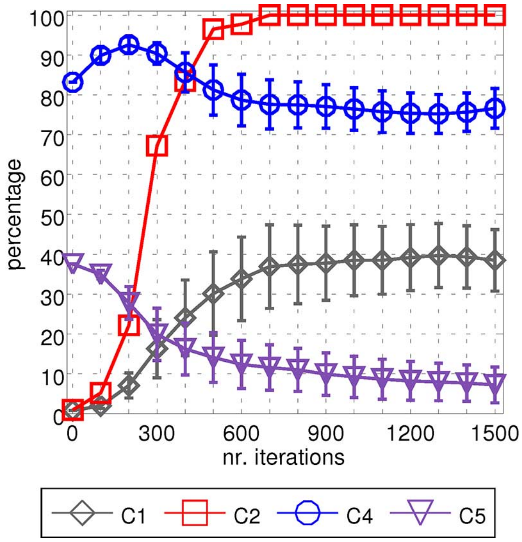


Fig. 12. Performance criteria from the experiments with the two AscTec Pelicans: current coverage $C_1(t)$, cumulative coverage $C_2(t)$, sensor quality $C_4(t)$, and risk $C_5(t)$ with respect to the number of iterations $t$.

## V. Discussion

This paper developed PARCov, a reactive motion-planning approach for persistent surveillance of risk-sensitive areas by a team of UAVs. PARCov relied on simple interactions among UAVs in order promote an emergent behavior that maximized coverage while reducing the risk and maintaining high sensor data quality. Scalability was achieved by separating motion planning in $xy$ from determining the optimal altitude for each quadcopter. Experiments in simulation and real AscTec Pelican quadcopters demonstrated the ability of the approach to provide persistent surveillance of risk-sensitive areas. While adding more quadcopters improves coverage it is also more costly. PARCov shows that it can obtain persistent coverage even with smaller teams. In future work, we will investigate the tradeoff between the number of quadcopters and operational cost. From a theoretical perspective, it would be interesting to analyze and modify the approach to achieve optimality while still remaining computationally efficient. We will also investigate the applicability of multi-objective optimization approaches which may allow incorporating additional objectives such as optimizing energy. Power management becomes a critical issue when conducting large-scale surveillance. In future work, we will also enhance the approach to track moving targets.

## Acknowledgment

The authors thank N. Sydney for valuable insights during this work and T. Apker for helping with the physical experiments. This work was performed at the Naval Research Laboratory. The views, positions and conclusions expressed herein reflect only the authors opinions and expressly do not reflect those of the U.S. Department of Defense, Office of Naval Research, or the Naval Research Laboratory.

## References

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[2] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Diego, CA, 1994, pp. 3310–3317.

[3] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 635–646, Aug. 2010.

[4] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *Int. J. Robot. Res.*, vol. 33, no. 3, pp. 401–425, 2014.

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, 2002, vol. 2, pp. 1327–1332.

[6] G. Hitz, A. Gotovos, F. Pomerleau, M.-E. Garneau, C. Pradalier, A. Krause, and R. Y. Siegwart, "Fully autonomous focused exploration for robotic environmental monitoring," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 2658–2664.

[7] I. Maza and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*. New York, NY, USA: Springer, 2007, pp. 221–230.

[8] Y. Jin, A. A. Minai, and M. M. Polycarpou, "Cooperative real-time search and task allocation in UAV teams," in *Proc. IEEE Conf. Decision Control*, Maui, HI, USA, 2003, vol. 1, pp. 7–12.

[9] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-UAV context," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, 2004, vol. 4, pp. 3622–3627.

[10] D. Zhu, H. Huang, and S. X. Yang, "Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 504–514, 2013.

[11] T. Apker, S.-Y. Liu, D. Sofge, and J. K. Hendrick, "Application of grazing-inspired guidance laws to autonomous information gathering," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Chicago, IL, USA, 2014, pp. 3828–3833.

[12] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. K. Hedrick, "An overview of emerging results in cooperative UAV control," in *Proc. IEEE Conf. Decision and Control*, Bahamas, 2004, vol. 1, pp. 602–607.

[13] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *J. Intell. Robot. Syst.*, vol. 57, no. 1–4, pp. 65–100, 2010.

[14] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun*, Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA, USA: MIT Press, 2005.

[15] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.

[16] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerosp. Sci. Technol.*, vol. 16, no. 1, pp. 47–55, 2012.

[17] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[18] M. Shanmugavel, A. Tsourdos, B. White, and R. Żbikowski, "Co-operative path planning of multiple UAVs using dubins paths with clothoid arcs," *Control Eng. Practice*, vol. 18, no. 9, pp. 1084–1092, 2010.

[19] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Trans. Syst., Man, Cybern., Part A: Syst. Humans*, vol. 42, no. 2, pp. 511–526, Mar. 2012.

[20] H. Ergezer and K. Leblebicioğlu, "3D path planning for multiple UAVs for maximum information collection," *J. Intell. Robot. Syst.*, vol. 73, no. 1–4, pp. 737–762, 2014.

[21] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[22] S. M. LaValle, "Motion planning: The essentials," *IEEE Robot. Autom. Mag.*, vol. 18, no. 1, pp. 79–89, 2011.

[23] E. Plaku, "Robot motion planning with dynamics as hybrid search," in *Proc. AAAI Conf. Artif. Intell.*, Bellevue, WA, USA, 2013, pp. 1415–1421.

[24] M. J. Kuhlman, P. Svec, K. N. Kaipa, D. Sofge, and S. K. Gupta, "Physics-aware informative coverage planning for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 4741–4746.

[25] R. W. Beard and T. W. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Proc. IEEE Conf. Decision Control*, Maui, HI, USA, 2003, vol. 1, pp. 25–30.

[26] P. Cheng, J. Keller, and V. Kumar, "Time-optimal UAV trajectory planning for 3d urban structure coverage," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Nice, France, 2008, pp. 2750–2757.

[27] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 410–426, Apr. 2012.

[28] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 947–961, Apr. 2013.

[29] V. A. Huynh, J. J. Enright, and E. Frazzoli, "Persistent patrol with limited-range on-board sensors," in *Proc. IEEE Conf. Decision Control*, Atlanta, GA, USA, 2010, pp. 7661–7668.

[30] P. R. Chandler, M. Pachter, and S. Rasmussen, "UAV cooperative control," in *Proc. IEEE Amer. Control Conf.*, Arlington, VA, USA, 2001, vol. 1, pp. 50–55.

[31] S. S. Baek, H. Kwon, J. A. Yoder, and D. Pack, "Optimal path planning of a target-following fixed-wing UAV using sequential decision processes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Tokyo, Japan, 2013, pp. 2955–2962.

[32] F. Lin, X. Dong, B. M. Chen, K.-Y. Lum, and T. H. Lee, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1038–1049, Feb. 2012.

[33] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying UAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, San Francisco, CA, USA, 2011, pp. 4929–4934.

[34] N. Sydney, D. A. Paley, and D. Sofge, "Physics-inspired robotic motion planning for cooperative Bayesian target detection," in *Proc. Robotics: Sci. Syst. Workshop on Distrib. Control and Estimation for Robotic Vehicle Networks*, Berkeley, CA, USA, 2014, .

[35] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Peer-to-Peer Systems*. New York, NY, USA: Springer, 2002, pp. 53–65.

[36] A. Wallar, E. Plaku, and D. A. Sofge, "A planner for autonomuos risk-sensitive coverage (PARCOV) by a team of unmanned aerial vehicles," in *Proc. IEEE Symp. Swarm Intell.*, Orlando, FL, USA, 2014, pp. 283–289.

[37] M. Srikanth, K. Bala, and F. Durand, "Computational RIM illumination of dynamic subjects using aerial robots," Computers and Graphics 2015. [Online]. Available: http://newsoffice.mit.edu/2014/drone-lighting-0711

[38] A. Fournier, D. S. Fussell, and L. C. Carpenter, "Computer rendering of stochastic models," *ACM Commun.*, vol. 25, no. 6, pp. 371–384, 1982.

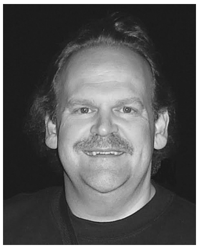
[39] L. Murphy and P. Newman, "Risky planning: Path planning over costmaps with a probabilistically bounded speed-accuracy tradeoff," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, 2011, pp. 3727–3732.

**Alex Wallar** received the B.Sc. degree from the School of Computer Science, University of St Andrews, Saint Andrews, Fife, U.K. He is currently working towards the Ph.D. degree at the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, and a student contractor with the Distributed Autonomous Systems Group, Naval Research Laboratory, Washington, DC, USA.

His research focuses on reactive path planning for ground and aerial vehicles.

**Erion Plaku** received the Ph.D. degree from Rice University, Houston, TX, USA.

He is an Assistant Professor with the Department of Electrical Engineering and Computer Science, The Catholic University of America, Washington, DC, USA. He was a Postdoctoral Fellow at the Laboratory for Computational Sensing and Robotics, The Johns Hopkins University, Baltimore, MD, USA. His research focuses on motion planning for mobile ground, underwater, and aerial vehicles.



**Donald A. Sofge** is a Computer Scientist with the Naval Research Laboratory, Washington, DC, USA, where he leads the Distributed Autonomous Systems Group. He has extensive expertise in the application of artificial intelligence, machine learning, machine vision (and other forms of artificial perception), planning and control theory to robotic systems. His research primarily focuses on the development of algorithms and techniques for controlling cooperative teams of autonomous systems.